Title: Apparatus and Method for Handling Electronic Mail

FIELD OF THE INVENTION

This invention relates to network communication systems and, in particular, to a system and method for handling incoming electronic mail messages.

BACKGROUND OF THE INVENTION

Denial of service attacks, including mail flooding, are common problems affecting the security of a mail transfer agent used in the management of electronic mail (e-mail). In the present state of the art, one possible response is to reject all incoming e-mail during a mail flooding or denial of service attack incident. However, such a response may be undesirable as e-mail from a legitimate sender can be rejected along with the mail flood from a problem sender. The present state of the art does not provide a method of distinguishing between the problem sender and the legitimate sender during periods of high e-mail activity.

What is needed is a method for responding to incoming e-mail which selectively rejects e-mail from a problem sender and selectively accepts e-mail from a legitimate sender during the onset of mail flooding or denial of service attack.

SUMMARY OF THE INVENTION

The present invention results from the observation that e-mail senders initiating undesirable activities can be identified and tracked by utilizing a penalty count filter module, integrated into the front end of a mail transfer agent in a communication device, so that subsequent e-mail disruptions to the receiving device can be mitigated. The penalty count filter module determines the disposition of incoming e-mail on the basis of current communication system resource usage and penalty counts assigned for cumulative undesirable sender activity. System resource usage may be determined by the number of concurrent TCP connections being maintained, and the undesirable sender activity, which is tracked using a behavior

trace table, may include sending a large number of e-mails or using a relatively large amount of TCP connection time. The penalty count filter module operates in a plurality of states, including a normal state, a selective-rejection state, and a random-rejection state. In the selective rejection state, e-mail from senders having a penalty count are randomly rejected, and e-mail from senders without a penalty count are accepted. In the random-rejection state, all e-mail from senders having a penalty count is rejected, and e-mail from senders without a penalty count are randomly rejected. The rejection rates can be increased or decreased in response to a detected increase or decrease in the usage of system resources.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention description below refers to the accompanying drawings, of which:

Fig. 1 is a diagrammatical representation of a mail transfer agent including a penalty count filter module for determining the disposition of incoming electronic mail;

Fig. 2 is a flow diagram describing the sequence of operations performed by the penalty count filter module of Fig. 1;

Fig. 3 is a data structure diagram of a penalty count table resident in the penalty count filter module of Fig. 1;

Fig. 4 is a flow diagram describing in greater detail the operation of the penalty count filter module, as represented by the step of determining filter module status in the flow diagram of Fig. 2;

Fig. 5 is a flow diagram describing in greater detail the operation of the penalty count filter module, as represented by the step of processing e-mail in accordance with sender and filter module status in the flow diagram of Fig. 2;

Express Mail Label No. EV339950545US

Fig. 6 is a diagrammatical illustration of a communications network including the penalty count rejection table of Fig. 3 resident in a mail transfer agent and a behavior trace table resident in a workstation;

Fig. 7 is a data structure diagram of the behavior trace table of Fig. 6;

Fig. 8 is a data structure diagram of a merged penalty count table which includes data from the penalty count table of Fig. 3 and the behavior trace table of Fig. 7; and

Fig. 9 is a flow diagram describing an alternative sequence of operations performed by the penalty count filter module of Fig. 1.

Express Mail Label No. EV339950205US

DETAILED DESCRIPTION OF AN ILLUSTRATIVE EMBODIMENT

There is shown in Fig. 1 a functional block diagram of a mail transfer agent 10, adapted for receiving an incoming e-mail message 11 via a communications network (not shown) such as the Internet. The mail transfer agent 10 can be integrated with a communication appliance, such as a personal computer or a workstation, for example. The mail transfer agent 10 includes a penalty count filter module (PCFM) 20 integrated into a front end 15 of the mail transfer agent 10, an incoming message queue 13, and a forwarding daemon 18. In a preferred embodiment, the front end 15 functions in accordance with Simple Mail Transfer Protocol (SMTP) as is well known in the relevant art.

The incoming e-mail message 11 is provided to an accept/reject filter 23 which determines whether to save the incoming e-mail message 11 to the incoming message queue 13 as an accepted e-mail message 16, or reject the incoming e-mail message 11 and transmit a transient negative completion reply 19 to the originator of the message. The accept/reject determination is made by the accept/reject filter 23 on the basis of a sender penalty count status 27 and a PCFM state 29. As explained in

greater detail below, a sender identifier 21 is obtained from the incoming e-mail message 11 and is used to determine the sender penalty count status 27.

The sender may be identified by using a reverse Domain Name Service (DNS) verification to ascertain the IP address of the originator of the incoming e-mail message 11. Under circumstances in which the envelope address of the sender is not available, such as during a TCP timeout attack, the peer IP address of the TCP connection can be used as the sender identifier 21. The sender penalty count status module 27 maintains a constantly-updated list of senders associated with undesirable e-mail activity. Such undesirable activity may include, for example, sending large numbers of e-mails, sending e-mails of relatively large sizes, using too much TCP connection time, or causing a TCP timeout.

The system resource usage status 25, which is updated whenever a TCP connection is established, is used to determine the PCFM state 29, as described in greater detail below. The system resource usage status 25 provides a value for the usage or capacity status of one or more system resources related to the processing of incoming messages, including the incoming e-mail message 11, such as disk space occupied by the incoming message queue 13, the number of e-mail files in the incoming message queue 13, or the number of concurrent TCP connections being maintained.

Operation of the mail transfer agent 10 can be described with additional reference to the flow diagram of Fig. 2, in which the mail transfer agent 10 completes a TCP connection, at step 31, and receives the incoming e-mail message 11. The PCFM state 29 is determined, at step 33, and the identity of the sender of the e-mail message 11 is determined, at step 35. The PCFM state 29 is determined as a function of: i) the current PCFM state, ii) the time period for which the penalty count filter module 20 has remained in the current PCFM state, and iii) the current system

resource usage status 25. Determination of the PCFM state 29 is not dependent on the identity of the sender of the e-mail message 11.

Using a process described in greater detail below, the accept/reject filter 23 determines, at step 37, either that the incoming e-mail message 11 is to be transmitted to the addressee as a forwarded e-mail message 17, or that the incoming e-mail message 11 is to be rejected and an optional transient negative completion reply 19 is to be transmitted to the sender, stating that the incoming e-mail message 11 has been rejected. The determination of accepting or rejecting the e-mail message 11 is made based on the current PCFM state 29 and the identity of the sender of the e-mail message 11. When the current incoming e-mail message 11 has been processed, the mail transfer agent 10 updates a bahavior trace table 160 (shown in Fig. 7, below), at step 41. In a preferred embodiment, the activity of each e-mail sender is tracked and updated by means of the behavior trace table 160. It should be understood that not all e-mail senders listed in the behavior trace table 160 will subsequently produce sufficient undesirable behavior to acquire a penalty count status.

The sender penalty count status module 27 determines if the incoming e-mail message 11 exhibits undesirable activity, at decision block 43, and if so, the sender penalty count status module 27 creates a new listing for the sender identifier 21 in a penalty count table 50 (shown in Fig. 3) with an appropriate sender penalty count value, at step 45. Or, if a listing already exists for the sender identifier 21, the corresponding sender penalty count value is updated. For normal e-mail activity in which no undesirable activity is detected, no change is made to the sender entry in the penalty count table 50. For a sender not listed in the penalty count table 50, the penalty count value is taken to be zero. In one embodiment of the present method, the activity of the sender of the e-mail message 11 is monitored after the TCP connection, established at step 31, has been terminated. Undesirable activity related to the current e-mail message 11 is noted, and the appropriate penalty count assessed. Operation of

the mail transfer agent 10 then proceeds to step 31, at which the next TCP connection is established.

Derivation of Penalty Count Status

The function of the penalty count table 50 can be described with reference to the data structure diagram in Fig. 3. The penalty count table 50 includes a plurality of records, represented by records 51, 53, 55, and 59. A semaphore feature 57, or a similar lock/unlock facility, is associated with each table entry for the purpose of synchronization when, for example, two or more processors are used to process the incoming e-mail message 11. The structure of the records 53, 55, and 59 are similar to that of the record 51 which includes a sender identifier (SID) value 61, a cumulative penalty count (PNCT) value 63, and a timestamp (TS) value 65.

In a preferred embodiment, the process of looking up sender identifier values can be optimized by using keys computed from the SIDs to hash the penalty count table 50. In a table with 256 entries, for example, the least significant byte of a corresponding sender identifier value can be used as the hash key. Rehashing can be minimized by a configuration in which each entry of the hash table points to a linked list, where the records are stored in the link list.

The cumulative penalty count value 63 is a time-dependent parameter which is updated in accordance with the behavior of the sender identified by the corresponding SID value 61. The timestamp value 65 records the time T_{TS} at which the cumulative penalty count value 63 was most recently calculated. The timestamp value 65 also provides for determining when the record 51 becomes out-of-date and should be removed from the penalty count table 50. In a preferred embodiment, the record 51 is removed after a retention period τ_{PCNT} of approximately 2^{19} seconds (i.e., about six days). The cumulative penalty count (PCNT) value 63 is preferably derived using the equation,

$$PCNT = \min(\kappa, \alpha + \varphi), \qquad (1$$

where κ is a pre-established maximum value for PCNT, α is an activity penalty count charged to the sender for the current undesirable activity, and φ is a previous penalty count value which was determined from the recent past history, if any, of undesirable activity produced by the sender. In a preferred embodiment, κ is set to 128.

The value for the activity penalty count α can be an integer specified by the system administrator, and may have a different value for different types of undesirable activities. For example, an activity penalty count of six may be assessed for sending a large number of e-mails exceeding a pre-established maximum quantity, an activity penalty count of four may be assessed for sending one or more e-mails exceeding a pre-established cumulative file size, and another activity penalty count may be assessed for incurring a TCP connection time exceeding a predetermined threshold. The activity penalty count is additive such that a sender can be assessed an activity penalty count of ten for exceeding both the maximum quantity and file size, for example.

The process of deriving the cumulative penalty count value 63 begins with the occurrence of an initial undesirable activity, for which a first activity penalty count of α_1 is charged to the sender identified by the SID value 61. As described above, the timestamp value 65 records the time (denoted as T_1 in the following example) of the occurrence of the current (i.e., the first) undesirable activity. Accordingly,

$$PCNT(T_1)=\min(\kappa, \alpha_1)=\alpha_1, \qquad (2$$

since $\varphi = 0$ where there has occurred only the first undesirable activity.

If the sender identified by the SID value 61 produces a subsequent (i.e., the second) undesirable activity at a time T_2, a second activity penalty count α_2 is assigned. If T_2 lies within the retention period τ_{PCNT}, the previous cumulative penalty count (PCNT) 63 is updated to the value $PCNT(T_2)$ using the expression,

$$PCNT(T_2)=\min(\kappa, \alpha_2+\varphi_2), \tag{3}$$

where,

$$\varphi_2=\left[PCNT(T_1)\cdot\left(1-\frac{T_2-T_1}{\tau_{PCNT}}\right)\right], \tag{4}$$

to give,

$$PCNT(T_2)=\alpha_2+\alpha_1\left(1-\frac{T_2-T_1}{\tau_{PCNT}}\right). \tag{5}$$

Note that, if T_2 occurs after the retention period τ_{PCNT} following T_1, then $\varphi_2=0$.

In general, the n[th] cumulative penalty count 63, updated at time T_n can be determined using the expression,

$$PCNT(T_n)=\min(\kappa, \alpha_{n-1}+\varphi_n), \tag{6}$$

where,

$$\varphi_n=\left[PCNT(T_{n-1})\cdot\left(1-\frac{T_n-T_{n-1}}{\tau_{PCNT}}\right)\right]. \tag{7}$$

Determination of PCFM State

In a preferred embodiment, the penalty count filter module 20 operates in one of at least three states: a 'normal' state, a 'selective-rejection' state, and a 'random-rejection' state. When the penalty count filter module 20 is operating in the 'normal' state, the mail transfer agent 10 accepts all valid incoming e-mail message 11 from any senders for transmittal to the intended addressees. As the system resources required to handle incoming e-mail volume increase and a greater demand is placed on the mail transfer agent 10, operation of the penalty count filter module 20 will change from the 'normal' state to either the 'selective rejection' state or the 'random rejection' state. Subsequently, if the e-mail volume returns to normal levels, operation of the penalty count filter module 20 eventually reverts to 'normal.'

If initially in the 'normal' state, operation of the penalty count filter module 20 changes from the 'normal' state to the 'selective-rejection' state if the system resources have increased beyond a first pre-established, 'selective-rejection' watermark. The selective-rejection watermark is reached, for example, when the disk space of the incoming message queue exceeds a predetermined disk-space threshold, or when the number of concurrent TCP connections exceeds a predetermined connection number. In alternative embodiments, other such criteria can be used, as specified by the system administrator, to define additional watermarks and operational states.

If initially operating in the 'random-rejection' state, the penalty count filter module 20 will remain in the 'random-rejection' state for at least a period of time denoted as a time-to-stay (TTS) interval τ_{TTS}. After the time-to-stay interval τ_{TTS} has passed, the penalty count filter module 20 may revert to either 'selective-rejection' operation or 'normal' operation, depending on the system resource usage status file 25. Alternatively, if initially operating in the 'selective-rejection' state, the penalty count filter module 20 will continue in the 'selective-rejection' operation for the time-to-stay interval τ_{TTS} and then revert to the 'normal' state, except when e-mail volume increases and operation of the penalty count filter module 20 is changed to the 'random-rejection' state. In a preferred embodiment, the time-to-stay interval τ_{TTS} is approximately ten minutes.

The application of the PCFM state 29 to the process of managing the incoming e-mail message 11 can be explained with reference to the flow diagram of Fig. 4, which provides a more detailed description of step 33 in Fig. 2. From step 31, the system resource usage status 25 is determined, at step 71. A query is made as to whether the penalty count filter module 20 is in the selective-rejection state, at decision block 73. If the response is 'yes,' operation proceeds to decision block 85. If the response is 'no' in decision block 73, a subsequent query is made as to whether

the penalty count filter module 20 is in the random-rejection state, at decision block 75. If the response is 'yes,' operation proceeds to decision block 115. If the response is 'no' in decision block 75, a query is made as to whether the system resource (SYSRES) has exceeded the selective-rejection watermark, at decision block 77. If the selective-rejection watermark has been exceeded, the time-to-stay is initialized to the time-to-stay interval τ_{TTS}, a resource usage factor f is set to an initial value, and the time-to-check is initialized to a time-to-check interval τ_{CHK}, at step 81. In a preferred embodiment, the time-to-check interval τ_{CHK} is approximately three seconds. The penalty count filter module 20 then changes to the selective-rejection state, at step 83. If the selective-rejection watermark has not been exceeded, at decision block 77, the penalty count filter module 20 remains in the normal state, at step 79.

If the penalty count filter module 20 is determined to be in the selective-rejection state, at decision block 73 (above), operation proceeds to decision block 85 at which a query is made as to whether the system resource has exceeded the random-rejection watermark. If the random-rejection watermark has been exceeded, the time-to-stay is initialized to the time-to-stay interval τ_{TTS}, a rejection factor R_f is set to an initial value, and the time-to-check is initialized to a time-to-check interval τ_{CHK}, at step 87. The rejection factor R_f has a value assigned by the system administrator and may have an initial value of two, for example. The penalty count filter module 20 changes to the random-rejection state, at step 89, and operation proceeds to step 35.

If the random-rejection watermark has not been exceeded, at decision block 85, a query is made as to whether the time-to-stay has expired, at decision block 91. If the time-to-stay has expired, a query is made as to whether the system resource has exceeded the selective-rejection watermark, at decision block 93. If the selective-rejection watermark has not been exceeded, the penalty count filter module 20 changes to the normal state, at step 95, and operation proceeds to step 35. If the

selective-rejection watermark has been exceeded, at decision block 93, the time-to-stay is updated, at step 97, the penalty count filter module 20 remains in the selective-rejection state, at step 99, and operation proceeds to step 35.

If the time-to-stay has not expired, at decision block 91, a query is made as to whether the time-to-check has expired, at decision block 101. If the time-to-check has not expired, the penalty count filter module 20 remains in the selective-rejection state, at step 99, and operation proceeds to step 35. If the time-to-check has expired, at decision block 101, a query is made as to whether the system resource has exceeded the selective-rejection watermark, at decision block 103. If the selective-rejection watermark has been exceeded, a resource usage factor f (defined below) is increased, at step 105, and the time-to-check is updated, at step 109. In one preferred embodiment, the resource usage factor f is doubled when the system resource exceeds the selective-rejection watermark. If the selective-rejection watermark has not been exceeded, at decision block 103, the resource usage factor f is decreased, at step 107, and the time-to-check is updated, at step 109. In another preferred embodiment, the resource usage factor f is decreased by a factor of two if the selective-rejection watermark has not been exceeded. After the time-to-check has been updated, at step 109, the penalty count filter module 20 remains in the selective-rejection state, at step 111, and operation proceeds to step 35.

If the penalty count filter module 20 is determined to be in the random-rejection state, at decision block 75 (above), operation proceeds to decision block 115 at which a query is made as to whether the time-to-stay has expired. If the time-to-stay has expired, a query is made as to whether the system resource has exceeded the random-rejection watermark, at decision block 117. If the random-rejection watermark has not been exceeded, the time-to-stay is initialized to the time-to-stay interval τ_{TTS}, the resource usage factor f is set to an initial value, and the time-to-check is initialized to the time-to-check interval τ_{CHK}, at step 119. Subsequently, the

penalty count filter module 20 changes to the selective-rejection state, at step 121, and operation proceeds to step 35. If the random-rejection watermark has been exceeded, at decision block 117, the time-to-stay is updated, at step 123, the penalty count filter module 20 remains in the random-rejection state, at step 125, and operation proceeds to step 35.

If the time-to-stay has not expired, at decision block 115, a query is made as to whether the time-to-check has expired, at decision block 127. If the time-to-check has not expired, the penalty count filter module 20 remains in the random-rejection state, at step 125, and operation proceeds to step 35. If the time-to-check has expired, at decision block 127, a query is made as to whether the system resource has exceeded the random-rejection watermark, at decision block 129. If the random-rejection watermark has been exceeded, the rejection factor R_f is increased, at step 131, and the time-to-check is updated, at step 135. In a preferred embodiment, the rejection factor R_f is doubled when the system resource exceeds the random-rejection watermark. If the selective-rejection watermark has not been exceeded, at decision block 129, the rejection factor R_f is decreased, at step 133, and the time-to-check is updated, at step 135. In one preferred embodiment, the rejection factor R_f is decreased by a factor of two if the random-rejection watermark has not been exceeded. After the time-to-check has been updated, at step 135, the penalty count filter module 20 remains in the random-rejection state, at step 137, and operation proceeds to step 35.

Processing E-mail

The operation of processing e-mail in accordance with sender status and the penalty count filter module state, at step 39 of Fig. 2, is shown in greater detail in the flow diagram of Fig. 5, in which a query is made, at decision block 201, as to whether the penalty count filter module 20 is in the selective-rejection state. If the response is 'yes,' a query is made as to whether the current value of the cumulative penalty count

φ is greater than zero, at decision block 203. If the response is 'no,' the incoming e-mail message 11 is accepted, at step 171, and sent to the addressee as forwarded e-mail 17. If the response is 'yes,' in decision block 203, a random number R_h is generated, at step 205, where $1 \leq R_h \leq \kappa$.

A query is then made, at decision block 207, as to whether the random number R_h is greater than the product of the resource usage factor f and the current value of the cumulative penalty count φ. If the response is 'yes,' the incoming e-mail message 11 is accepted, at step 171, and forwarded to the addressee. Operation returns to step 41 where the bahavior trace table 160 is updated. If the response is 'no,' at decision block 207, the incoming e-mail message 11 is rejected, at step 209, the penalty count filter module 20 returns the transient negative completion reply 19 to the sender, and operation returns to step 41.

If the response is 'no,' at decision block 201, a query is made as to whether the penalty count filter module 20 is in the random-rejection state, at decision block 211. If the response is 'no,' the incoming e-mail message 11 is accepted, at step 171, and sent to the addressee as forwarded e-mail 17. If the response is 'yes,' at decision block 211, a query is made as to whether the current value of the cumulative penalty count φ is greater than zero, at decision block 213. If the response is 'yes,' at decision block 213, the incoming e-mail message 11 is rejected, at step 219, and the penalty count filter module 20 returns the transient negative completion reply 19 to the sender. If the response is 'no,' at decision block 213, a random number R_g is generated, at step 215, where $1 \leq R_g \leq \kappa$.

A query is made, at decision block 217, as to whether the random number R_g is greater than the rejection factor R_f. If the response is 'yes,' the incoming e-mail message 11 is accepted, at step 171, and forwarded to the addressee. If the response is 'no,' at decision block 217, the incoming e-mail message 11 is rejected, at step 219, the penalty count filter module 20 returns the transient negative completion reply 19

to the sender, and operation returns to step 41 where the bahavior trace table 160 is updated.

As described above, the resource usage factor f is used at step 207 to determine whether a particular incoming e-mail message 11 is to be randomly returned to the sender or transmitted to the intended addressee. The resource usage factor f is derived from the average cumulative penalty count $\overline{P}$, where

$$\overline{P} = \frac{\sum_{i=1}^{m} PCNT_i}{m}, \tag{8}$$

and where $PCNT_i$ is the penalty count assessed to the ith sender identifier, of the m sender identifiers listed in the penalty count table 50. The resource usage factor f is given by the equation,

$$f = \frac{\kappa}{2\overline{P}}. \tag{9}$$

As described above, the penalty count filter module 20 then generates the random number R_h, which is compared to the product of the resource usage factor f and the current value of the cumulative penalty count φ. If $R_h > f \cdot \varphi$, the incoming e-mail message 11 is accepted by the penalty count filter module 20; if $R_h \leq f \cdot \varphi$, the incoming e-mail message 11 is rejected and the penalty count filter module 20 issues the transient negative completion reply 19 to the corresponding sender.

Behavior Trace Table

In a preferred embodiment, the behavior trace table 160 is included in a workstation 150 which is connected with the mail transfer agent 10 by means of a communication network 151, such as a LAN or WAN, as shown in Fig. 6. Information included in the behavior trace table 160, which tabulates e-mail activities of the sender, is used by the sender penalty count status 27 to update the cumulative penalty count φ of an e-mail sender in the penalty count table 50.

The behavior trace table 160 includes a plurality of records, or behavior values, represented by records 161, 163, 165 and 169, shown in Fig. 7. A semaphore feature 167, or a similar lock/unlock facility, is associated with each table entry. The structure of the records 163, 165, and 169 are similar to that of the record 161 which includes a sender identifier (SID) value 171, a cumulative e-mail count (NN) 173 of the sender e-mails, a total e-mail size (SZ) 175 of the e-mail files, a total TCP connection utilization time (UT) 177, and a timestamp (TS) value 179 which is a record of the most recent e-mail received from the sender identified by the SID value 171.

The process of looking up sender identifier values can be optimized by using keys computed from the SIDs to hash the behavior trace table 160, using a method similar to that for looking up values in the penalty count table 50. Rehashing can be minimized in the behavior trace table 160 by having each entry of the hash table point to a linked list, where the records are stored in the link list. Each record 161, 163, 167, and 169 in the behavior trace table 160 has a retention period of τ_{BHT}, after which the out-of-date record is removed when the corresponding table entry is visited and the record is traversed. In a preferred embodiment, the retention period τ_{BHT} is approximately five seconds.

When the incoming e-mail message 11 corresponds to the sender identified by the sender identifier value 171, the cumulative e-mail count 173, the cumulative total size 175, and the cumulative total TCP connection utilization time 177 are updated using current and previous values, and where the previous values are reduced by a decay factor d. The updated cumulative e-mail value (NN)is given by the equation,

$$NN = 1 + d \cdot NN(T_{TS}), \qquad (10$$

where $NN(T_{TS})$ is the prior or most recent previous e-mail count value, previously obtained at the timestamp time (T_{TS}).

The updated cumulative total size (SZ) is given by the equation,

$$SZ = SZ(T_{UD}) + d \cdot SZ(T_{TS}), \quad (11$$

where $SZ(T_{UD})$ is the additional e-mail size obtained at the time of updating the record and $SZ(T_{TS})$ is the prior or most recent previous e-mail size. Similarly, the updated cumulative connection utilization time (UT) is given by the equation,

$$UT = UT(T_{UD}) + d \cdot UT(T_{TS}), \quad (12$$

where $UT(T_{UD})$ is the connection utilization time determined at the time of updating the record and $UT(T_{TS})$ is the prior or most recent previous connection utilization time. After the cumulative e-mail count 173, the cumulative total size 175, and the cumulative total TCP connection utilization time 177 have been updated, the timestamp (T_{TS}) is reset to the most recent time of update. The decay factor d in equations 10 to 12 is given by,

$$d = 1 - \min\left(1, \frac{T_{TC} - T_{TS}}{\tau_{BHT}}\right). \quad (13$$

If any of the cumulative e-mail count 173, the cumulative total size 175, or the cumulative total TCP connection utilization time 177 exceeds a predefined threshold or watermark, as discussed above, the sender corresponding to the sender identifier value 171 is assessed an appropriate penalty count. For incoming e-mail message 11 sent by a sender not currently listed in the behavior trace table 160, a new entry is created with initial values of one, $SZ(T_{TS})$, and $UT(T_{TS})$ entered into the behavior trace table 160 for the cumulative e-mail count 173, the cumulative total size 175, and the cumulative total TCP connection utilization time 177, respectively.

In yet another embodiment, the mail transfer agent 10 includes a merged penalty count table 180, shown in Fig. 8, which includes a plurality of records having entries similar to that shown for a record 181. The record 181 results from merging the data in the record 51 and the record 161, and includes a sender identifier (SID) value 183, a cumulative penalty count (PCNT) value 185, a cumulative e-mail count (NN) 187 of the sender e-mails, a total e-mail size (SZ) 189 of the e-mail files, a total

TCP connection utilization time (UT) 191, and a timestamp (TS) value 193. As can be appreciated by one skilled in the relevant art, a behavior trace table 160 is not required in the workstation 150 if the mail transfer agent 10 includes the merged penalty count table 180 in place of the penalty count table 50.

In an alternative embodiment, shown in the flow diagram of Fig. 9, the incoming e-mail 11 is processed in accordance with sender status. The mail transfer agent 10 completes a TCP connection, at step 231, and receives the incoming e-mail message 11. The identity of the sender of the e-mail message 11 is determined, at step 233. The accept/reject filter 23 determines, at step 235, either that the incoming e-mail message 11 is to be transmitted to the addressee or that the incoming e-mail message 11 is to be rejected. The determination of accepting or rejecting the e-mail message 11 is made based on the identity of the sender. When the current incoming e-mail message 11 has been processed, the behavior trace table 160 is updated, at step 239, and the sender penalty count status module 27 determines if the incoming e-mail message 11 exhibits undesirable activity, at decision block 241. If so, the sender penalty count status module 27 creates a new listing for the sender identifier 21 in the penalty count table 50 with an appropriate sender penalty count value, at step 243. Or, if a listing already exists for the sender identifier 21, the corresponding sender penalty count value is updated. For normal e-mail activity in which no undesirable activity is detected, no change is made to the sender entry in the penalty count table 50. For a sender not listed in the penalty count table 50, the penalty count value is taken to be zero. Operation of the mail transfer agent 10 then proceeds to step 231, at which the next TCP connection is established.

While the invention has been described with reference to particular embodiments, it will be understood that the present invention is by no means limited to the particular constructions and methods herein disclosed and/or shown in the

drawings, but also comprises any modifications or equivalents within the scope of the claims.

What is claimed is:

Page 6 of 6